Laboratorio Nro. 1
Recursión

Objetivos

1. Resolver problemas usando recursión
2. Corroborar los resultados teóricos obtenidos con la notación asintótica O

Consideraciones iniciales

Leer la Guía

Antes de comenzar a resolver el presente laboratorio, leer la ***"Guía Metodológica para la realización y entrega de laboratorios de Estructura de Datos y Algoritmos"*** que les orientará sobre los requisitos de entrega para este y todos los laboratorios, las rúbricas de calificación, el desarrollo de procedimientos, entre otros aspectos importantes.

Registrar Reclamos

En caso de tener **algún comentario** sobre la nota recibida en este u otro laboratorio, pueden **enviarlo** a través de http://bit.ly/2g4TTKf, el cual será atendido en la menor brevedad posible.

DOCENTE MAURICIO TORO BERMÚDEZ
Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627
Correo: mtorobe@eafit.edu.co

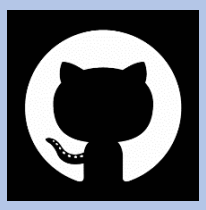
Traducción de Ejercicios

En el GitHub del docente, encontrarán la traducción al español de los enunciados de los Ejercicios en Línea.

Visualización de Calificaciones

A través de ***Eafit Interactiva*** encontrarán **un enlace** que les permitirá **ver un registro de las calificaciones** que **emite el docente** para cada taller de laboratorio y según las rubricas expuestas. ***Véase sección 3, numeral 3.8.***

GitHub



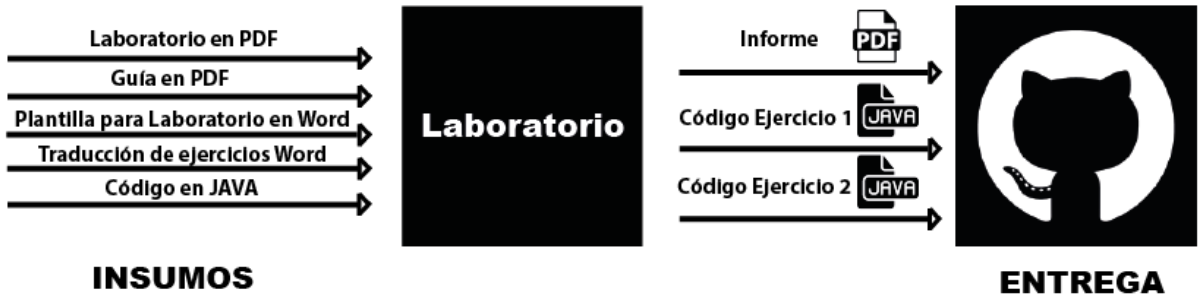
Crear un repositorio en su cuenta de GitHub con el nombre `st0245-eafit`. **2.** Crear una carpeta dentro de ese repositorio con el nombre `laboratorios`. **3.** Dentro de la carpeta `laboratorios`, crear una carpeta con nombre `lab01`. **4.** Dentro de la carpeta `lab01`, crear tres carpetas: `informe`, `codigo` y `ejercicioEnLinea`. **5.**Subir el informe pdf a la carpeta `infome`, el código del ejercicio 1 a la carpeta `codigo` y el código del ejercicio en línea a la carpeta `ejercicioEnLinea`. Así:

```
st0245-eafit
  laboratorios
      lab01
         informe
         codigo
         ejercicioEnLinea
      lab02
         ...
```

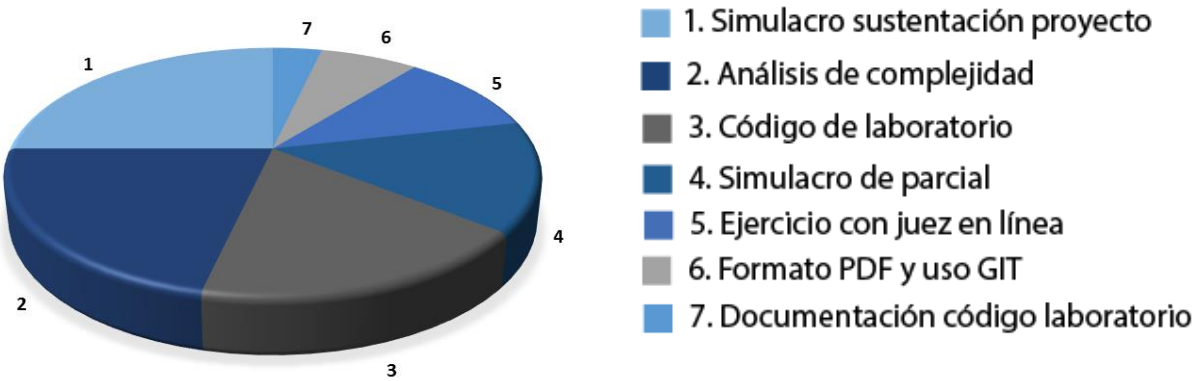
DOCENTE MAURICIO TORO BERMÚDEZ
Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627
Correo: mtorobe@eafit.edu.co

Intercambio de archivos

Los archivos que **ustedes deben entregar** al docente son: **un archivo PDF** con el informe de laboratorio usando la plantilla definida, y **dos códigos**, uno con la solución al numeral 1 y otro al numeral 2 del presente. Todo lo anterior se entrega en **GitHub**.



Porcentajes y criterios de evaluación para el laboratorio



DOCENTE MAURICIO TORO BERMÚDEZ
Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627
Correo: mtorobe@eafit.edu.co

Ejercicios a resolver

1. Códigos para entregar en GitHub:

En la vida real, la documentación del software hace parte de muchos estándares de calidad como CMMI e ISO/IEC 9126

Véase Guía ***en Sección 3, numeral 3.4***

*Código de laboratorio en **GitHub.*** Véase Guía en **Sección 4, numeral 4.24**

*Documentación en **HTML***

No se reciben** archivos en **.RAR** ni en **.ZIP

En la vida real, la serie de Fibonacci se aplica para modelar dinámica poblacional de ecosistemas, construcción de montículos y análisis de obras de arte y fotografías

1.1 Teniendo en cuenta lo anterior, extiendan el código existente para medir los tiempos de *Array Sum recursivo*, *Array Maximum recursivo* y *serie de Fibonacci recursivo* para arreglos generados aleatoriamente con diferentes tamaños.

PISTA: *Véase **Guía sección 4, numeral 4.6** "Cómo usar la escala logarítmica en Microsoft Excel".*

DOCENTE MAURICIO TORO BERMÚDEZ
Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627
Correo: mtorobe@eafit.edu.co

PISTA 2: Si todos los tiempos de un algoritmo dan más de 5 minutos, realice otra tabla, para ese algoritmo, tomando tiempos para arreglos de tamaño 1000, 10000 y 100000.

PISTA 3: Véase **Guía sección 4, numeral 4.4** "Cómo aumentar el tamaño del heap y del stack en Java"

PISTA 4: Véase **Guía sección 4, numeral 4.5** "Cómo visualizar el montículo (heap) y el stack, y el consumo total de memoria de Java"

NOTA: Los ejercicios del numeral 1 deben ser documentados en formato HTML. Véase *Guía en* ***Sección 4, numeral 4.1*** *"Cómo escribir la documentación HTML de un código usando JavaDoc"*

2) Ejercicios en línea sin documentación HTML en GitHub

GUÍA	Véase Guía en ***Sección 3, numeral 3.3***	HTML	**No entregar** documentación **HTML**
JAVA	*Entregar un archivo en* ***.JAVA***	PDF	***No se reciben*** *archivos en* ***.PDF***
	Resolver *los problemas de* ***CodingBat*** *usando* ***Recursión***		*Código del ejercicio en línea en* ***GitHub.*** Véase Guía en **Sección 4, numeral 4.24**

NOTA: Recuerden que, **si toman la respuesta de alguna fuente**, deben **referenciar** según el **tipo de cita** correspondiente**.** Véase *Guía en* ***Sección 4, numerales 4.16 y 4.17***

DOCENTE MAURICIO TORO BERMÚDEZ
Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627
Correo: mtorobe@eafit.edu.co

2.1 Resolver al menos 5 ejercicios del nivel *Recursion 1* de CodingBat: http://codingbat.com/java/Recursion-1

Error común

2.2 Resolver al menos 5 ejercicios del nivel *Recursion 2* de la página CodingBat: http://codingbat.com/java/Recursion-2

NOTA: No está permitido el ejercicio ***GroupSum***. Pero a continuación encontrarán algunos errores comunes que pueden ser aplicados con los otros ejercicios

PISTA: El algoritmo *GroupSum* falla porque al llamarse recursivamente con el parámetro `start` se queda en una recursión infinita

```
public boolean groupSum(int start, int[] nums, int target) {
  if (start >= nums.length) return target == 0;
  return groupSum(start+1, nums, target - nums[start])
        ||groupSum(start, nums, target );
}
```

PISTA 2: El algoritmo *GroupSum* falla porque al llamarse recursivament con el parámetro `start-1` se sale del arreglo cuando `start = 0`.

```
public boolean groupSum(int start, int[] nums, int target) {
  if (start >= nums.length) return target == 0;
  return groupSum(start+1, nums, target - nums[start])
        ||groupSum(start-1, nums, target );
}
```

PISTA 3: El algoritmo *GroupSum* falla porque al llamarse recursivamente con el parámetro `start` se sale del arreglo cuando `start = length-1`

```
public boolean groupSum(int start, int[] nums, int target) {
  if (start >= nums.length) return target == 0;
  return groupSum(start+1, nums, target - nums[start])
        ||groupSum(start+1, nums, target - nums[start - 1] );
```

2.3 Expliquen con sus propias palabras cómo funciona el ejercicio *GroupSum5*

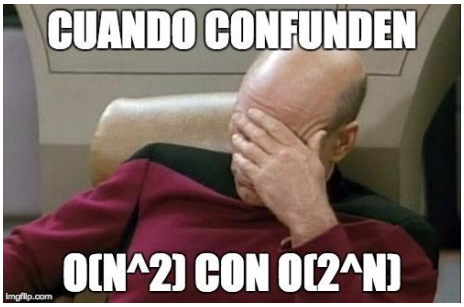
NOTA: Recuerden que debe explicar su implementación en el informe PDF

2.4 Calculen la complejidad de los Ejercicios en Línea de los numerales 2.1 y 2.2 y agréguenla al informe PDF

PISTA: Véase ***Guía en Sección 4, numeral 4.11*** *"Cómo escribir la complejidad de un ejercicio en línea"*

PISTA 2: Véase ***Guía en Sección 4, numeral 4.19 "****Ejemplos para calcular la complejidad de un ejercicio de CodingBat"*

Errores Comunes



2.5 Expliquen con sus palabras las variables (qué es 'n', qué es 'm', etc.) del cálculo de complejidad del ejercicio 2.4

3) Simulacro de preguntas de sustentación de Proyectos

GUÍA	Véase Guía en ***Sección 3, Numeral 3.5***	PDF	*Entregar informe de laboratorio* en **PDF**
	*Usen la **plantilla** para responder laboratorios*	ICONTEC	***No** apliquen **Normas Icontec** para esto*

DOCENTE MAURICIO TORO BERMÚDEZ
Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627
Correo: mtorobe@eafit.edu.co

3.1 De acuerdo a lo realizado en el numeral 1.1, completen la siguiente tabla con tiempos en milisegundos:

	N = 100.000	N = 1'000.000	N = 10'000.000	N = 100'000.000
R Array sum				
R Array maximum				
R Fibonacci				

En la vida real, las gráficas son una forma simple y concisa de representar resultados cuantitativos en Ingeniería de Sistemas

3.2 Grafiquen los tiempos que tomó en ejecutarse Array Sum, Array Maximum y *Fibonacci recursivo*, para entradas de diferentes tamaños. Si se demora más de un minuto la ejecución, cancela la ejecución y escriba en la tabla "más de 1 minuto".

Con *Fibonacci*, haga las pruebas con valores menores a 25 o 30. Grafiquen el Tamaño de la Entrada Vs. Tiempo de Ejecución. Utilice una escala logarítmica para poder graficar correctamente

PISTA: *Véase **Guía sección 4, numeral 4.6** "Cómo usar la escala logarítmica en Microsoft Excel".*

PISTA 2: Si todos los tiempos de un algoritmo dan más de 5 minutos realice otra tabla, para ese algoritmo, tomando tiempos para arreglos de tamaño 1000, 10000 y 100000.

3.3 ¿Qué concluyen respecto a los tiempos obtenidos en el laboratorio y los resultados teóricos?

3.4 ¿Qué aprendieron sobre *Stack Overflow*?

PISTA: http://bit.ly/1TRr3HL

3.5 ¿Cuál es el valor más grande qué pudo calcular para Fibonnacci? ¿Por qué? ¿Por qué no se puede ejecutar Fibonacci con 1 millón?

PISTA: http://bit.ly/2ix7rjl

3.6 ¿Cómo se puede hacer para calcular el Fibonacci de valores grandes?

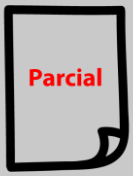
PISTA: http://bit.ly/2ix7rjl

3.7 ¿Qué concluyen sobre la complejidad de los problemas de *CodingBat Recursion 1* con respecto a los de *Recursion 2*?

4) Simulacro de Parcial en el informe PDF

PISTA: *Véase* ***Guía en Sección 4, Numeral 4.18*** "Respuestas del Quiz"

PISTA 2: Lean las diapositivas tituladas ***"Data Structures I: O Notation (recursion)"*** encontrarán la mayoría de las respuestas



Para este simulacro, agreguen ***sus respuestas*** **en el informe PDF.**

El ***día del Parcial no tendrán computador, JAVA o acceso a internet.*** Usen una hoja de papel para dar sus respuestas.

1. Pepito escribió un algoritmo que, dado un arreglo de enteros, decide si es posible escoger un subconjunto de esos enteros, de tal forma que la suma de los elementos de ese subconjunto sea igual a `target`. El parámetro `start` funciona como un contador y representa un índice en el arreglo de números `nums`.

```
01 public boolean SumaGrupo(int start,
                      int[] nums, int target) {
02  if (start >= nums.length) return target == 0;
03  return SumaGrupo(start + 1, nums,
                             target - nums[start])
04         || SumaGrupo(____,___,___);
05 }
```

¿Qué parámetros colocaría en el llamado recursivo de la línea 4 para que el programa funcione?

_____, _____, _____

2. Pepito escribió el siguiente código usando recursión

```
private int b(int[ ] a, int x,
              int low, int high) {
  if (low > high) return -1;
  int mid = (low + high)/2;
  if (a[mid] == x) return mid;
  else if (a[mid] < x)
```

DOCENTE MAURICIO TORO BERMÚDEZ
Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627
Correo: mtorobe@eafit.edu.co

```
        return b(a, x, mid+1, high);
    else
        return b(a, x, low, mid-1);
}
```

¿Cuál ecuación de recurrencia describe el comportamiento del algoritmo anterior para el peor de los casos?

a) $T(n)=T(n/2)+C$
b) $T(n)=2.T(n/2)+C$
c) $T(n)=2.T(n/2)+Cn$
d) $T(n)=T(n-1)+C$

3. Dayla y Kefo están aquí de nuevo. En esta vez han traído un juego muy interesante, en el cual Kefo, en primer lugar, escoge un numero n ($1 \leq n \leq 20$) y, en segundo lugar, escoge tres números a,b y c ($1 \leq a \leq 9, 1 \leq b \leq 9, 1 \leq c \leq 9$). Después, Kefo le entrega estos números a Dayla y Dayla le tiene que **decir a Kefo la cantidad máxima de números tomados de a, b y c (se puede tomar un número más de una vez) que al sumarlos dan el valor n**.

Como un ejemplo, si Kefo escoge n=14 y a=3,b=2,c=7. ¿Qué posibilidades hay de obtener 14 con a,b y c?

7+7=14	cantidad es 2
7+3+2+2=14	cantidad es 4
3+3+3+3+2=14	cantidad es 5
…	
2+2+2+2+2+2+2=14	cantidad es 7

La cantidad máxima de números es 7. Esta sería la respuesta que da Dayla a Kefo. Como Dayla es muy astuta, ha diseñado un algoritmo para determinar la cantidad máxima de números y quiere que le ayudes a terminar su código. Asuma que hay al menos una forma de sumar n usando los números a, b y c en diferentes cantidades, incluso si algunos de los números se suman 0 veces como sucede en el ejemplo anterior.

```
1 int solucionar (int n, int a, int b, int c) {
2   if (n == 0 || (n < a && n < b && n < c))
3      return 0;
4   int res = solucionar(_____________) + 1;
5   res = Math.max(_______,___________);
6   res = Math.max(_______,___________);
7   return res; }
```

3.1. Complete el espacio de la línea 04

_____

3.2. Complete los espacios de la línea 05

_____, _____

3.3. Complete los espacios de la línea 06

_____, _____

4. ¿Qué calcula el algoritmo `desconocido` y cuál es la complejidad asintótica en el peor de los casos del algoritmo `desconocido`?

```
01 public int desconocido(int[] a){
02    return aux(a, a.length-1); }
03
04 public int aux(int[] a, int n){
05    if(n < 1) return a[n];
06    else return a[n] + aux(a, n-1); }
```

a) La suma de los elementos del arreglo a y es
b) Ordena el arreglo a y es $O(n.log\ n)$
c) La suma de los elementos del arreglo `a` y es $O(1)$
d) El máximo valor de un arreglo `a` y es $O(n)$
e) La suma de los elementos del arreglo `a` y es $O(n)$

DOCENTE MAURICIO TORO BERMÚDEZ
Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627
Correo: mtorobe@eafit.edu.co

5. [Ejercicio Opcional] Lectura recomendada

"Quienes se preparan para el ejercicio de una profesión requieren la adquisición de competencias que necesariamente se sustentan en procesos comunicativos. Así cuando se entrevista a un ingeniero recién egresado para un empleo, una buena parte de sus posibilidades radica en su capacidad de comunicación; pero se ha observado que esta es una de sus principales debilidades…"

Tomado de http://bit.ly/2gJKzJD

Véase Guía en ***Sección 3, numeral 3.6 y 4.20*** *de la Guía Metodológica, "Lectura recomendada" y "Ejemplo para realización de actividades de las Lecturas Recomendadas*", respectivamente

Posterior a la lectura del texto ***"Narasimha Karumanchi, Data Structures and Algorithms made easy in Java, (2nd edition), 2011. Chapter 3: Recursion and Backtracking"*** realicen las siguientes actividades que les permitirán sumar puntos adicionales:

a) Escriban un resumen de la lectura que tenga una longitud de 100 a 150 palabras

PISTA: En el siguiente enlace, unos consejos de cómo hacer un buen resumen http://bit.ly/2knU3Pv

PISTA 2: Aquí le explican cómo contar el número de palabras en Microsoft Word

b) Hagan un mapa conceptual que destaque los principales elementos teóricos.

PISTA: Para que hagan el mapa conceptual se recomiendan herramientas como las que encuentran en https://cacoo.com/ o https://www.mindmup.com/#m:new-a-1437527273469

NOTA: Si desean otra lectura, consideren la siguiente: ***"John Hopcroft et al., Fundamentos de Algoritmia. Capítulo 3: Notación Asintótica. Páginas 98 – 106. 1983".*** que encuentran en biblioteca

NOTA 2: Estas respuestas también deben incluirlas en el informe PDF

6. [Ejercicio Opcional] Trabajo en Equipo y Progreso Gradual

El trabajo en equipo es una exigencia actual del mercado. "Mientras algunos medios retratan la programación como un trabajo solitario, la realidad es que requiere de mucha comunicación y trabajo con otros. Si trabajas para una compañía, serás parte de un equipo de desarrollo y esperarán que te comuniques y trabajes bien con otras personas"

Tomado de http://bit.ly/1B6hUDp

GUÍA *Véase Guía en **Sección 3, numeral 3.7** y **Sección 4, numerales 4.21, 4.22** y **4.23** de la Guía Metodológica*

a) Entreguen copia de todas las actas de reunión usando el tablero Kanban, con fecha, hora e integrantes que participaron

PISTA: *Véase* ***Guía en Sección 4, Numeral 4.21*** "*Ejemplo de cómo hacer actas de trabajo en equipo usando Tablero Kanban"*

b) Entreguen el reporte de *git*, *svn* o *mercuria*l con los cambios en el código y quién hizo cada cambio, con fecha, hora e integrantes que participaron

PISTA: ***Véase*** *Guía en Sección 4, Numeral 4.23* "**Cómo generar el historial de cambios en el código de un repositorio que está en svn"**

c) Entreguen el reporte de cambios del informe de laboratorio que se genera *Google docs* o herramientas similares

PISTA: ***Véase*** *Guía en Sección 4, Numeral 4.22* "***Cómo ver el historial de revisión de un archivo en Google Docs*"**

NOTA: Estas respuestas también deben incluirlas en el informe PDF

Resumen de ejercicios a resolver

1.1 Medir los tiempos de Array Sum recursivo, Array Maximum recursivo y serie de Fibonacci recursivo

2.1 Resolver al menos 5 ejercicios del nivel Recursion 1 de CodingBat: http://codingbat.com/java/Recursion-1

2.2 Resolver al menos 5 ejercicios del nivel Recursion 2 de la página CodingBat: http://codingbat.com/java/Recursion-2

2.3 Expliquen con sus propias palabras cómo funciona el ejercicio GroupSum5

2.4 Calculen la complejidad de los Ejercicios en Línea de los numerales 2.1 y 2.2 y agréguenla al informe PDF

2.5 Expliquen con sus palabras las variables (qué es 'n', qué es 'm', etc.) del cálculo de complejidad del numeral anterior

3.1 De acuerdo a lo realizado en el numeral 1.1, completen la tabla con tiempos en milisegundos

3.2 Grafiquen los tiempos que tomó en ejecutarse Array Sum, Array Maximum y Fibonacci recursivo, para entradas de diferentes tamaños.

3.3 ¿Qué concluyen respecto a los tiempos obtenidos en el laboratorio y los resultados teóricos?

3.4 ¿Qué aprendieron sobre Stack Overflow?

3.5 Cuál es el valor más grande qué pudo calcular para Fibonnacci? ¿Por qué? ¿Por qué no se puede ejecutar Fibonacci con 1 millón?

3.6 ¿Cómo se puede hacer para calcular el Fibonacci de valores grandes?

3.7 ¿Qué concluyen sobre la complejidad de los problemas de CodingBat Recursion 1 con respecto a los de Recursion 2?

4. Simulacro de Parcial

5. Lectura recomendada **[Ejercicio Opcional]**

6. Trabajo en Equipo y Progreso Gradual **[Ejercicio Opcional]**